

**PATENT APPLICATION TRANSMITTAL LETTER**  
(Small Entity)

Docket No.  
BRN-10202/03

TO THE ASSISTANT COMMISSIONER FOR PATENTS

Transmitted herewith for filing under 35 U.S.C. 111 and 37 C.F.R. 1.53 is the patent application of:

**Caroline Baser et al**

For: **ELECTRONIC COMMERCE METHOD AND APPARATUS**

Enclosed are:

- ☒ Certificate of Mailing with Express Mail Mailing Label No. EJ335012152US
- ☒ Nine (9) informal sheets of drawings.
- ☐ A certified copy of a \_\_\_\_\_ application.
- ☒ Declaration ☐ Signed. ☒ Unsigned.
- ☒ Power of Attorney
- ☐ Information Disclosure Statement
- ☐ Preliminary Amendment
- ☒ Verified Statement(s) to Establish Small Entity Status Under 37 C.F.R. 1.9 and 1.27.
- ☐ Other:

**CLAIMS AS FILED**

For	#Filed	#Allowed	#Extra	Rate	Fee
Total Claims	32	- 20 =	12	x \$9.00	\$108.00
Indep. Claims	3	- 3 =	0	x \$41.00	\$0.00
Multiple Dependent Claims (check if applicable) <input type="checkbox"/>					\$0.00
BASIC FEE					\$380.00
TOTAL FILING FEE					\$488.00

- ☒ A check in the amount of \$488.00 to cover the filing fee is enclosed.
- ☒ The Commissioner is hereby authorized to charge and credit Deposit Account No. 07-1180 as described below. A duplicate copy of this sheet is enclosed.
  - ☐ Charge the amount of \_\_\_\_\_ as filing fee.
  - ☒ Credit any overpayment.
  - ☒ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
  - ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).

Dated: Feb. 24, 1999

*Signature*

**John G. Posa**  
Reg. No. 37,424  
Gifford, Krass, Groh et al  
280 N. Old Woodward Ave., Suite 400  
Birmingham, MI 48009  
734/913-9300

cc:

## **ELECTRONIC COMMERCE METHODS AND APPARATUS**

### Reference to Related Applications

This application claims priority of U.S. provisional patent application Serial No. 60/075,872, filed February 25, 1998, the entire contents of which are incorporated herein by  
5 reference.

### Field of the Invention

This invention relates generally to electronic commerce and, in particular, to an object-oriented, distributed architecture providing a suite of applications for  
10 implementing prepaid e-commerce in internet, intranet, and extranet environments.

### Background of the Invention

Although a precise definition has yet to be adopted, electronic commerce or e-commerce, as it is called, generally  
15 means the process of conducting business transactions via networked computers, whether through direct connections or over the internet. Regardless of the definition, e-commerce includes various business-to-business transactions, including many on-line retail and financial processes.

20 The trend toward e-commerce began some 25 years ago, when larger corporations began transacting business with

subsidiaries and suppliers over private networks. Recently, e-commerce has become more synonymous with business carried out over the internet, through webs sites, for example, to sell goods, services, and information to consumers.

5 Reliability, security and privacy remain concerns with regard to conducting business over the internet. Being packet-switched, internet messages typically move through various computers before arriving at a final destination, raising the potential for unauthorized interception. Existing  
10 architectures that depend upon internet protocols or e-mail communications therefore generally employ encryption schemes to enhance security. Other approaches address such problems through direct links to commercial banking networks, which often require users to open and maintain an account.

15 Generally speaking, most existing approaches to e-commerce rely on encryption involving the distribution of public or private encryption keys, which, in turn, requires measures to ensure that the keys are not lost or stolen.

Existing techniques also present problems in terms  
20 of network comparability. Telephony platforms based upon voice communications use switched networks, whereas the internet relies on packet routing. Whereas the switched network is based upon establishing a physical link, the internet routes packets from node-to-node to establish a  
25 communication. There is concern among internet users that the

source and destination addresses are, themselves, not secure and subject to hacking activities.

Prior-art approaches to e-commerce generally use a hybrid configuration involving both switched and packet  
5 routing. A dial-up connection is used to gain access to the internet, for example. This hybrid approach generally does not yield the best of both worlds, however, since encryption keys must still be used with source and destination addresses subject to unauthorized tampering. The need still remains,  
10 therefore, for a pure e-commerce solution, preferably one which permits different kinds of transactions to occur through a consistent architecture.

#### Summary of the Invention

The subject invention resides in electronic commerce  
15 systems and processes, particularly with respect to the processing of pre-paid transactions. The invention is specifically directed to an object-oriented, distributed architecture providing a suite of applications in support of electronic commerce through web service platforms on the  
20 internet or other points of sale, including intranet and extranet environments. The invention is not limited by the size of the transactions, and takes into account micropayments of the type which might otherwise be economically impractical through credit-card purchases.

A method of performing a pre-paid electronic-commerce transaction includes the steps of receiving a service request from a user, and creating a transaction instance. Information relating to the user's PIN and remaining balance  
5 are retrieved to determine whether or not the transaction can take place given the user's remaining balance and, if the user's PIN is sufficiently funded, the transaction proceeds, rendering the requested service. An unrated service data record is returned, and the price of the goods or services is  
10 calculated based upon the service data record. The PIN balance is updated, and a transaction data record is generated.

Five device components are preferably employed to complete a transaction, thus enabling any transaction server  
15 (TxS<sup>™</sup>) to be local or remote. A preferred system for carrying out electronic commerce in accordance with the invention includes the following components, the operations of which are coordinated by a transaction manager: an input device, an account device, a rating device, a service device, and an  
20 output device.

The input device is employed for receiving a request for a service from an end-user through a business application, forming part of an internet web page, for example, or through the use of a virtual device called the Payment Portal<sup>™</sup>. The  
25 Payment Portal may be viewed as an expandable widget/icon

which resides independently on any web browser/page. When selected the widget/icon is activated on the WEB page being viewed, allowing the end user to select the payment method, enter into a secure mode and then, if prepaid is selected, 5 enter a debit account number (PIN) and a password in the fields provided for access to the prepaid account.

The account device performs account-management functions, including PIN verification operations. The rating device is used to calculate the purchase price associated with 10 the service to be provided, as when purchasing time for multi-user games, downloading newspaper articles, etc. The service device actually provides the service. The output device for maintains transaction data record (TDR) queues accessible through a business application.

15 Through transaction shipment, an end user of the invention is not limited by geographic or national boundaries. Such roaming enables the end user to travel outside of a home location to an access point such as a web service platform on the internet or intranet/extranet/point-of-sale terminal in a 20 country or region different from the one in which they normally receive service.

A typical roaming situation involves two TxS<sup>™</sup> devices. The first device, identified as a foreign TxS, is the place where the end user initiates the transaction. The 25 second device, identified as the home TxS, holds the business

information for the pre-paid account. It is the home device which is associated with a TxS that holds the PIN business information.

The foreign TxS receives the end-user request and renders the service, once the PIN balance has been checked. The home TxS is requested by the foreign TxS to retrieve the PIN balance, perform the rating, and update the account. The nature of the distributed architecture enables roaming to be performed without any PIN data replication, thereby eliminating data inaccuracy and consistency issues.

The use of transaction shipment is not restricted to a typical roaming situation, but also applies to more complex roaming situations. For example, in a 3-TxS roaming situation, TxS#1 captures the request (ID) and services the requests (SD); TxS#2 holds the universal account device (AD) and the output device (OD); and TxS#3 holds the rating modules (RD).

#### Brief Description of the Drawings

FIGURE 1 is a drawing of a four-layer architecture according to the invention;

FIGURE 2 is a drawing of a business model according to the invention including five components coordinated by a transaction manager;

FIGURE 3 illustrates different transaction steps

associated with a pre-paid electronic commerce example;

FIGURE 4 illustrates the transaction of Figure 3 in the form of a flow diagram;

FIGURE 5 illustrates a typical roaming situation  
5 involves two transaction server devices;

FIGURE 6 illustrates a transaction shipment according to the invention;

FIGURE 7 depicts a roaming situation utilizing three transaction server devices;

10 FIGURE 8 illustrates an example using a pre-paid service from an HTML Web Page;

FIGURE 9 is a diagram which shows how an end user may enter order information in through an HTML page;

FIGURE 10 is a portion of an HTML page which  
15 contains a form whose action property invokes CorbaCgiServlet.doGet;

FIGURE 11 provides a simplified code sample of CorbaCgiServlet;

FIGURE 12 illustrates important transaction steps  
20 associated with a web service example;

FIGURE 13 shows how, if more sophisticated features are required on the client side, the service HTML page can include references to embedded Java applets;

FIGURE 14 is a UAD Entity Relationship Diagram based  
25 on a simple, two-entity model;



FIGURE 15 represents a JAVA application and JAVA installation according to the invention;

FIGURE 16 illustrates X/Open and OTS interfaces with respect to a credit/debit example according to the invention;

5           FIGURE 17 is a UML diagram which depicts a transaction inheritance tree;

FIGURE 18 is a UML diagram which depicts a transaction data record;

10           FIGURE 19 is a UML diagram which depicts the transaction server and devices according to the invention;

FIGURE 20 is a UML diagram which depicts requests and transaction creation; and

15           FIGURE 21 is a UML diagram which depicts object interaction for the transaction shipments that occur in the typical roaming situation.

#### Detailed Description of the Invention

This invention resides in electronic commerce systems and processes, particularly with respect to the processing of pre-paid transactions. The invention is specifically directed to an object-oriented, distributed architecture providing a suite of applications in support of true electronic commerce, as might be implemented through various wide- and local-area network service platforms or other points of sale (POS), including smart devices such as a

PDA (Personal Digital Assistant), or virtual devices such as the Payment Portal<sup>™</sup> described in further detail below.

Although specific examples will make reference to the internet, one of skill in the art of distributed transaction processing will recognize that the invention is equally applicable to intranet and extranet environments. Nor is the invention limited by the size of the transactions, and takes into account micropayments of the type which might otherwise be economically impractical through credit-card purchases.

In a preferred embodiment the invention uses a consistent four-layer architecture as shown in Figure 1. Broadly, the architecture integrates a business application suite to an enterprise node called the transaction server (TxStm), in such a way that transactions may be executed in a fully distributed mode. That is, a transaction initiated on one node may involve other nodes in order to be completed.

The top layer of the architecture is the business application level which pulls transaction records from the TxS and performs back-office operations. The TxS engine layer manages pre-paid distributed transactions. The TxS engine is free of vendor-specific features, and has an open architecture, in that pre-paid accounts can be initially set up by an existing debit platform vendor's software, then later used for other kinds of pre-paid services through the TxS.

The device encapsulation layer enables a vendor platform or a server/device to interface with the TxS. The TxS interface defines a set of operations allowing distributed transactions. This layer is platform specific, in that  
5 interfaces must be written every time one integrates with a new type of service platform. However, since all WEB servers comply with standard interface specifications, this layer has already been implemented. The bottom layer is the service platform layer, which may function as a web server, point of  
10 sale, PDA, or other smart device.

The TxS engine defines interfaces for CORBA (Common Object Request Broker Architecture) services, and each service has a corresponding set of operations. Once a service has been made available, any of its operations can be requested  
15 over the network. Integrating a service platform with the TxS also implements its related interfaces. With respect to prepaid telephony, the TxS does not execute any debit functions or call processing. The TxS performs all debit processing steps as a pure e-commerce server solution for  
20 prepaid payments over internet/intranet/extranet.

As shown in Figure 2, the TxS business model consists of five components whose coordination is ensured by a transaction manager (TM). These components are:

1. The input device through which end-user  
25 requests are received, as through a web page or virtual device

forming part of a web-enabled service.

2. The account device which is responsible for carrying out lock, read and update PIN operations.

3. The rating device, which calculates purchase  
5 price of goods and services for the end-user.

4. The service device, which renders the actual service, such as connecting multiple parties through a TCP/IP connection.

5. The output device, which maintains multiple  
10 transaction data record (TDR) queues available for the business application. The TxS engine defines interfaces for creating CORBA services, such that each service owns a set of operations. Once a service has been made available, any operation can be requested across the network.

15 Payment Portal<sup>™</sup>

In terms of the input device, the Payment Portal is one the techniques that the TxS may use for making network payments for products or services with a prepaid payment method using a debit account (or with a post-paid payment  
20 method associated with a credit card number). The Payment Portal may be viewed as an expandable widget/icon which resides independently on any web browser/page. When selected, the widget/icon will expand on the WEB page being viewed, allowing the end user to select the payment method, enter into

a secure mode and then, if prepaid is selected, enter a debit account number (PIN) and a password in the fields provided.

The web page requesting payment will pass account device location and rate device location information to the  
5 Payment Portal. If a prepaid method is being used, the authorization of the account can be executed. If a purchase is authorized, the account decrementation can be accomplished at the account device specified. The final balance after the purchase will be shown in the Payment Portal.

10 A history of payments can be provided to the end user. In case of an estimate of the item's price, the rate device specified can be used to calculate the price without a purchase being made or without sufficient funds being available. The results are shown to the user in the Payment  
15 Portal's user interface.

Another use of the Payment Portal will be to top off (i.e., reload) any prepaid account balance at the specified account device with a credit card or through a home banking application where the end user can move some amount of money  
20 from a bank or credit card account to the prepaid account.

#### PRE-PAID E-COMMERCE TRANSACTIONS

By way of a review, the transaction server or TxS is an e-commerce server that can be used for various types of pre-paid services, including pre-paid services on the

Internet. With its account service, the TxS provides debit account management services, including account authorization, debiting and updating. Figures 3 and 4 illustrate the different transaction steps associated with a pre-paid  
5 electronic commerce example. Referring first to Figure 3, the primary steps are as follows:

1. Incoming request
2. The transaction manager creates a transaction instance
- 10 3. PIN authorization, lock and remaining balance retrieval
4. The TM requests the RD to determine whether or not the transaction can take place, given the remaining balance.
- 15 5. If the PIN is sufficiently funded, the transaction manager requests the service device to proceed with the transaction.
6. Service platform renders the requested service (for example, placing an order).
- 20 7. The service device returns a raw (not rated) service data record (SDR).
8. The raw SDR is handed over to the rating device in order to calculate the cost of the transaction (ie, the purchase price of the goods or services).
- 25 9. The transaction manager requests the account

device to update the PIN balance and unlock the PIN.

10. A TDR is queued to the output device

11. The business application retrieves the TDR.

Figure 4 illustrates the transaction in the form of  
5 a flow diagram.

#### TYPICAL ROAMING SITUATION (TRS)

As discussed above, five device components are preferably employed to complete a transaction, thereby enabling any TxS device to be local or remote. Roaming  
10 provides the ability for any end-user to travel outside of their home location by a local access point (for example, a POS) in a country or region different from the one in which they normally receive service. An end-user is not limited by geographic or national boundaries. Roaming creates a global  
15 system for end users.

Figures 5 illustrates a typical roaming situation involves two TxS devices. The first device, identified as a foreign TxS, is the place where the end user initiates the transaction. The second device, identified as the home TxS,  
20 holds the business information for the pre-paid account. This platform is associated with a TxS that holds the PIN business information. The foreign TxS receives the end user request and renders the service once the PIN balance has been checked. The home TxS is requested by the foreign TxS to retrieve the

PIN balance, to perform the rating and update the account. All the transaction steps are executed. The rating device and the account device operations are executed on a remote TxS. Note that the distributed architecture enables roaming to be  
5 performed without any PIN data replication, thereby eliminating data inaccuracy and consistency issues.

Two transaction managers are involved in the roaming situation just described. The transaction keeps running on one node as long as transaction steps can be executed locally.  
10 The transaction ships to another TxS node if a transaction step cannot be executed locally.

#### TRANSACTION SHIPMENT FOR TYPICAL ROAMING SITUATION

It is important that it should only take two messages back and forth to carry out a TRS transaction. This  
15 is achieved with transaction shipment, as shown in Figure 6. In order to minimize the number of CORBA messages, a transaction keeps running on a TxS node as long as its local devices can service the request. The transaction ships to another TxS as soon as an operation (i.e., a transaction step)  
20 cannot be performed locally. Transaction shipment is of course not restricted to TRS transactions. It actually applies to all roaming situations. Figure 7 depicts a 3-TxS roaming situation, wherein:

1. TxS#1 captures the request (ID) and services the



requests (SD),

2. TxS#2 holds the universal account device (AD)  
and the output device (OD), and

3. TxS#3 holds the rating modules (RD).

5

#### WEB SERVICE PLATFORM EXAMPLE

The following section describes an example using a  
pre-paid service from an HTML Web Page, as depicted in Figure  
8. This application would be useful, for example, for placing  
any kind of order from a WEB browser. This embodiment  
10 implements an input device that allows the end-user to enter  
a pre-paid transaction request from an HTML page. The primary  
goal of this integration is to deliver the end-user request to  
the TxS. Being a true e-commerce transaction, the account  
device need not be specific, allowing a basic universal  
15 account device to be used in this implementation. The account  
device may even be an account device previously used for  
another kind of service.

A specific rating device must be implemented, since  
parameters entered from the WEB page will be used to calculate  
20 the price of the goods or services purchased. For example, in  
a pizza ordering service, the end user chooses a pizza size  
and the toppings. These are quantifiable parameters used for  
calculating the pizza price. Discounts, club points,  
promotions and other sales methods can also be part of the

rating model used. The service device takes care of placing the actual order. It might consist of writing a record in a database, or for example, sending a fax to the fulfillment center.

- 5           In this particular example, the input device may include a CORBA servlet that can be requested from any network node. The end user enters the order information in the HTML page, as shown in Figure 9. When the end user submits its request, the HTTP Servlet DoGet method is invoked. The DoGet  
10 method locates and requests the CORBA Server.

The order entered on the Web page is eventually stored in a database by the service device. The orders placed in a database can be processed by an external application. The implementation comprises the following elements:

- 15 Web Server:

Any Web Server.

Input Device:

This component is an Input Device Implementation created from the MTG CORBA interface **tmg.engine.InputDevice**.

- 20 OrderServer:

CORBA service object associated with the Input Device. It is used by the Web Server to send an End User Request to the TxS.

The Input Device creates the CORBA Server instance. The Order Server is a registered CORBA service whose name is "OrderServer" and that was created from the CORBA interface `tools.web.CorbaCgiInterface`.

5 `CorbaCgiServlet`:

`Tools.web.CorbaCgiServlet` extends `javax.servlet.http.HttpServlet`. Its `doGet` method is called when the user submits their page to the HTTP Server. The `CorbaCgiServlet` is a CORBA client of the Order Server.

10 The HTML page contains a form whose "action" property invokes `CorbaCgiServlet.doGet`. It also has a property which gives the name of the OrderServer CORBA service, as shown in Figure 10. The `CorbaCgiServlet`, shown in the paragraph of Figure 11, provides a simplified code sample of `CorbaCgiServlet`. The `doGet` method retrieves the  
15 `AdsServer` name from the HTTP Request parameters and locates the corresponding CORBA service. The `doGet` method uses the servlet output stream to return a dynamically built HTML page.

Transactions Steps (Figure 12)

20 0 The end-user submits the page to the Web Server. The HTTP request contains all the data entered by the end user (Prefix, PIN number, Order Description). The Do Get

method of the Servlet is called.

- 1     The DoGet Method locates the CORBA Order Servlet associated with the input device and calls its "exec" method with the end user request as a parameter. The order  
5     servlet places the request in the input device queue. The "exec" method is synchronous, and waits until the end of the transaction.
- 2     The transaction manager reads the ID queue and creates a transaction instance.
- 10    3     PIN authorization, lock and remaining balance retrieval
- 4     The TM requests the rating device to calculate the price of the goods or services purchased.
- 5     If the PIN is sufficiently funded, the  
15    transaction manager requests the service device to proceed with the transaction.
- 6     Create a new order in the orders database.
- 7     The order has been created and the service device returns a service data record
- 20    8     Unlike telephony, the rating operation for ordering a physical item here is a void one, since the purchase price is known before the transaction has ended. However, for services that are metered, such as, interactive games and chat encounters, the rating will be done in  
25    real-time as the transaction is on-going.

9 The transaction manager requests the account device to update the PIN balance and unlock the PIN. A TDR is created and placed in one of the transaction manager queue.

10 The order server "exec" dynamically creates an  
5 HTML response page depending on the transaction outcome and then returns it to the CorbaCgiServlet.

11 The CorbaCgiServlet prints the dynamically created page into its output stream. The page is then displayed in the end user's Web browser.

#### 10 Alternative Architecture (Figure 13)

The Web Service architecture described above uses plain HTML pages that do not contain downloadable applets. A servlet is used for establishing communication with the TxS and this servlet is a client of the CORBA Order Server. If  
15 more sophisticated features are required on the Client side, the service HTML page can include references to embedded Java applets. In this architecture, shown in Figure 13, the applet is the Client of the CORBA Order Server.

The Java client typically interacts with the server as  
20 follows:

1. Web Browser downloads HTML page that includes JAVA applets references.
2. Web Browser retrieves JAVA applet from HTTP Server.

3. The Java virtual machine embedded in Web Browser loads the applet and starts it.

4. The applet is now running on the client and invokes the Order Server using CORBA.

#### 5 Universal Account Device (Figure 14)

The Universal Account Device, or UAD, is a PIN database that can be used for any pre-paid service. The UAD implementation preferably uses Oracle 8.0. The UAD Entity Relationship Diagram is based on a very simple, two-entity model, as shown in Figure 14. Simultaneous pre-paid transactions using the same account is possible as long as business rules are well defined. The introduction of such business rules allowing simultaneous use can be described as Soft Locking versus Hard Locking that locks out anyone else when the pre-paid account is being used.

#### Output Device

The output device is the counterpart to the input device, in the sense that it is in charge of managing the TDR after transactions have taken place; namely, queuing, pull interface for the business application, and so forth. The output device also supports multiple transactions queues and multiple client TDR retrieval.

### JAVA Applications

As shown in Figure 15, JAVA applications preferably use the J'Express 3.0 Full Java installation product, which includes:

- 5 · The TransactionServer
- The OnLineDecisionSystem
- The RealTimeEngine

### Fault Tolerance

Fault tolerance is required for the TxS Trader and  
10 for each TxS node. Visibroker provides fault tolerance by starting instances on multiple hosts. If a client is connected to an object implementation, and the connection is lost, the Visibroker Smart Agent detects the loss and automatically reconnects the client to another instance.  
15 However, Trader and TxS are object implementations that maintain states, which means that a lost connection will not be transparent to the client. Visibroker provides events' handler mechanisms to bring the state of a replica implementation up to date.

### 20 Load Balancing

With Load Balancing, requests can be automatically routed to different servers so as to dynamically balancing the request load. Visibroker uses Smart Stubs to provide load

balancing (The smart stub invokes the least loaded of several equivalent remote objects).

#### Administration GUI

First stage is a simple administration GUI that  
5 enables an administrator to change device configurations  
without bringing down the TxS. This mainly includes add/  
remove programs/prefixes to and from a TxS.. At the present  
time, TxS devices are preferably loaded from a file at start  
up time and cannot be changed unless a TxS shutdown is  
10 performed. Second stage will include a slicker interface  
intended for commercial use (i.e., device graphical  
representations, additional information supervision  
information).

#### SNMP Agent

15 The SNMP agent (e.g., Java Advent) provides an  
interface to a manager application so that the TxS can be  
supervised and monitored. This interface preferably contains:

- A set of variables that facilitates monitoring (the  
manager can invoke set and get operations for these  
20 variables),
- A set of Trap messages (these messages can be sent  
to the manager upon detection of an abnormal  
condition or for example when a variable goes above



or beyond a pre-defined threshold).

### Counter Agent

The motivation behind the counter agent is a licensing policy based on transaction volume and number of occurrences. For example, a license fee might be paid for the first 10 million transactions, with upgrade fees being required if a predefined threshold is exceeded. A counter agent will manage transaction counting. If invoked, the counter agent will have the ability to shut down the TxS upon reaching a predetermined threshold limit.

### OTS Interfaces and X/OPEN Mapping

Figure 16 illustrates X/Open and OTS Interfaces with respect to a credit/debit example. The main OTS interfaces are:

1. Current is a CORBA pseudo object that makes it easy for clients to use OTS. Clients invoke begin and commit operations.

2. Coordinator is implemented by the transaction service. Recoverable objects use it to coordinate their participation in the transaction. Phase#1 occurs when the coordinator asks each participant to vote (prepare operation). If all participants agree, coordinator performs phase#2 (asks all participants to commit).

3. Resource: recoverable server object participating in a two-phase commit (Prepare operation is the vote)

#### Transaction Management and Transaction Shipping

The objective of transaction shipping is to minimize the number of CORBA messages in roaming situations. In a roaming situation, a transaction ships or travels from one TxS to another. Therefore, it globally consists of consecutive run sequences executed on different TxS devices. The global transaction can also be seen as an execution path: The same TxS may of course appear several times in an execution path but two adjacent TxS are necessarily different.

A run sequence consists of several operations. The outcome of one operation determines what the next operation is. The transaction ships if the next operation cannot be executed locally.

A TxS knows what its own capabilities are (they are loaded from the TxS devices file at start time or updated via the administration GUI application). A transaction thread can determine at any time from end user request properties if a local device can perform the next operation or if shipment needs to occur.

Each TxS caches the other TxS it establishes communication with. TxS-to-TxS communication mainly occurs for shipping transactions. When a TxS needs to ship a

transaction to another TxS, it first looks it up in its TxS cache. The alternative is:

1. The looked-up TxS is not found in the cache. It asks the trader the name of the TxS that can perform the operation, gets a reference to that TxS (CORBA bind), caches the reference and ships the transaction, and

2. The looked-up TxS is found in the cache. The TxS just uses the cached remote reference (with no preparatory ping!) and tries to ship the transaction. If the first shipping attempt fails, that means the cached reference is obsolete. The obsolete reference is removed from the cache and the TxS does as if the reference had not been found in the cache in the first place.

#### DESIGN DIAGRAMS

15 This section of the description provides further details of the technical design utilizing the UML notation. Class diagrams will first be presented, followed by a TRS interaction diagram for the shipments which occur in a typical roaming situation.

20

#### CLASS DIAGRAMS

##### Transaction Inheritance Tree (Figure 17)

The TxS is capable of dealing with different business

models. Each transaction sub-class matches a business model (e.g.: pre-paid, post-paid, and so forth), containing all the behaviors required for implementing that model. Each transaction is a sequence of operations whose execution order  
5 is coded in each subclass. Upon termination of an operation (or transaction step), the sub-class analyzes its outcome and decides what the next one will be and if a shipment needs to occur. The transaction super-class holds necessary information for general transaction management (transaction  
10 state, reference to the local TxS, next operation to be executed, etc.)

#### Transaction Data Record (Figure 18)

- The TDR aggregates all the information that pertains to a transaction. It consists of three data groups:
- 15       1. General: Identification, State, Begin and End Time Dates
  2. End User Request: Origin, Type (the kind of transaction that is requested: pre-paid, post-paid, etc... PIN, Prefix; and
  - 20       3. Service Data Request: Record return by the Service Device proceed call. Price, quantity, Additional properties specific to the service device.

As the transaction progresses and possibly travels around, the TDR builds up. It is completed when the last step

of the transaction has been executed. The shippable transaction class is only used for shipping. Its purpose is to make shipments as small as possible. One needs to be able to create a shippable transaction from a transaction (at  
5 shipment time) and vice versa (upon receipt of a shipped transaction). The shippable transaction only consists of the next step to be executed and the TDR.

#### TxS and Devices (Figure 19)

The TxS class is the TxS engine. It is associated with  
10 device implementations through the five interfaces: input device, account device, rating device, service device and output device. The role of the TxS devices is described elsewhere herein. The TxS engine manipulates its devices only through its interfaces and does not know about device  
15 implementations. Device implementation classes are chosen at start-up time. When a TxS starts up, it advertises its responsibilities to the TxS Trader. The TxS uses the TxS Trader to locate other TxSs across the network.

#### Requests and Transaction Creation (Figure 20)

20 When an end user request or a shipped transaction comes in, the TxS just hands it over to the transaction manager that is in charge of creating the corresponding transaction sub-class and starting a transaction thread.

The TxS uses the transaction manager through an interface.

The transaction manager uses the transaction creator through an interface. The transaction creator can carry out three  
5 different operations:

1. Turn an end user request into a transaction interface reference (a new request has come in)

2. Turn a shippable transaction into a transaction interface reference (a shipped transaction has come in)

10 3. Turn a transaction interface reference into a shippable transaction (a shipment needs to occur).

The transaction manager always manipulates transactions through the transaction interface and does not know anything about transaction subclasses. The only class that needs to  
15 know about transaction sub-classes is the transaction creator.

#### TRS Interaction

Figure 21 depicts object interaction for the transaction shipments that occur in the typical roaming situation. In this example, the transaction is initiated on the foreign TxS  
20 and first ships on to home TxS to execute the following transactions steps:

AD.pinStatus,

AD.pinBalanceAndLock, and

RD.maxQuantity.

BRN-10202/03  
92302sh

It then ships back to the Foreign TxS for proceeding with the service device. It ships again to Home TxS to execute:

RD.rate,  
AD.pinDebit, and  
5 OD.queueTdr.

We claim:

1. A method of performing a pre-paid electronic-  
2 commerce transaction for a user having a personal  
identification number (PIN), the method comprising the steps  
4 of:

receiving a request for goods or services from the user  
6 and creating a transaction instance;

retrieving account information relating to the user's  
8 PIN, including the user's remaining balance;

determining whether or not the transaction can take place  
10 as a function of the user's remaining balance;

proceeding with the transaction and servicing the request  
12 if the user's account is sufficiently funded;

generating an unrated service data record;

14 calculating the purchase price of the requested goods or  
services based upon the service data record;

16 updating user's remaining balance; and

generating or updating a transaction data record.

2. The method of claim 1, wherein the transaction  
2 occurs over the internet.

3. The method of claim 1, wherein the transaction  
2 occurs over an intranet.

4. The method of claim 1, wherein the transaction



2 occurs over an extranet.

5. The method of claim 1, wherein the request is  
2 received through a point-of-sale (POS) terminal.

6. The method of claim 1, further including the step .  
2 of:

denying further service requests when a predetermined  
4 threshold is reached, thereby enforcing usage limits.

7. The method of claim 1, wherein the input device  
2 forms part of a web page.

8. The method of claim 1, wherein the input device is  
2 a virtual device.

9. The method of claim 8, wherein the virtual device  
2 allows the user to:

select a pre-paid method of payment, and  
4 enter the user's PIN and password.

10. The method of claim 1, wherein the step of  
2 calculating the purchase price of the requested goods or  
services occurs in real time.

11. The method of claim 10, wherein the purchase price  
2 is a dollar or less.

12. The method of claim 1, wherein:  
2 the steps associated with receiving the request from the  
user and servicing the request are performed at a first  
4 location; and

one or more of the other steps are performed at one or  
6 more different locations.

13. The method of claim 12, wherein one of the different  
2 locations is associated with requesting a payment, and wherein  
that location passes accounting and rating information to the  
4 first location.

14. The method of claim 1, further including the step of  
2 providing the user with an estimated purchase price before a  
purchase is made.

15. The method of claim 1, further including the step of  
2 providing the user with a history of payments.

16. The method of claim 1, further including the step of  
2 allowing the user to move funds from a bank or credit card  
account to increase the remaining balance.

17. The method of claim 1, wherein the input device  
2 forms part of a personal digital assistant.

18. An architecture facilitating e-commerce  
2 transactions, comprising:

a) an input device for receiving a request for goods or  
4 services from a user having a personal identification number  
(PIN);

6 b) an account device for performing account-management  
functions, including account balance and PIN verification  
8 operations;

c) a rating device for calculating the price of the  
10 requested goods or services;

d) a service device for fulfilling the request;

12 e) an output device for maintaining one or more  
transaction data records; and

14 f) a transaction manager coordinating the operations of  
a) through e).

19. The architecture of claim 18, including transactions  
2 which occur over the internet.

20. The architecture of claim 18, including transactions  
2 which occur over an intranet.

21. The architecture of claim 18, including transactions  
2 which occur over an extranet.

22. The architecture of claim 18, including transactions  
2 which originate at a point-of-sale (POS) terminal.

23. The architecture of claim 18, wherein the input  
2 device is an internet web page.

24. The architecture of claim 23, wherein the virtual  
2 device allows the user to:  
select a pre-paid method of payment, and  
4 enter the user's PIN and password.

25. The architecture of claim 18, wherein at least the  
2 rating device and the service device support micropayments.

26. The architecture of claim 18, wherein the input  
2 device and the service device are disposed at a first  
location, and one or more of the other devices are disposed at  
4 one or more different locations.

27. The architecture of claim 18, wherein the rating  
2 device is operative to provide the user with an estimated

purchase price before a purchase is made.

28. The architecture of claim 18, wherein the input  
2 device is operative to provide the user with a history of  
payments.

29. The architecture of claim 18, wherein the output  
2 device supports multiple service data record queues.

30. The architecture of claim 18, wherein the account  
2 device is operative to move funds from a bank or credit card  
account to increase the remaining balance.

31. The architecture of claim 18, wherein the input  
2 device forms part of a personal digital assistant.

32. A method of performing a pre-paid electronic-  
2 commerce transaction, comprising the steps of:

- a) providing the architecture of claim 18;
- 4 b) receiving a request from a user through the input  
device;
- 6 c) creating a transaction instance through the  
transaction manager;
- 8 d) performing PIN authorization, lock and remaining  
balance retrieval through the account device;

BRN-10202/03  
92302sh

10           e)     invoking the rating device to determine whether or  
              not the transaction can take place given the user's remaining  
12 balance;

              f)     proceeding with the transaction through the service  
14 device if the PIN is sufficiently funded;

              g)     fulfilling the request if the PIN is sufficiently  
16 funded;

              h)     returning an unrated service data record through the  
18 service device;

              i)     calculating a purchase price using the rating  
20 device;

              j)     updating the PIN balance and unlocking the PIN using  
22 the account device; and

              k)     creating multiple TDR queues.

BRN-10202/03 92302sh

Abstract of the Disclosure

An object-oriented, distributed architecture provides a suite of applications supporting pre-paid electronic commerce on internet, intranet or extranet, including roaming transactions and transaction shipping involving more complex configurations. A service request is received from a user, creating a transaction instance. Information relating to the user's PIN and remaining balance are retrieved to determine whether or not the transaction can take place given the user's remaining balance and, if the user's PIN is sufficiently funded, the transaction proceeds, rendering the requested service. An unrated service data record is returned, and the end-user purchase price of the requested goods or services is calculated based upon the service data record. The PIN balance updated, and a transaction data record is generated as well. A typical roaming situation involves two transaction servers, called TxS devices. The first device, identified as a foreign TxS, is the place where the end user initiates the transaction. The second device, identified as the home TxS, holds the business information such as PIN funding for the pre-paid account. The use of transaction shipment is not restricted to a typical roaming situation, but also applies to more complex roaming situations. An input device is employed for receiving a request for a service from an end-user through a business application, forming part of an internet web page,

BRN-10202/03  
92302sh

for example, or through the use of a virtual Payment Portal<sup>™</sup>,  
which is manifest as an expandable widget/icon.






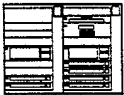
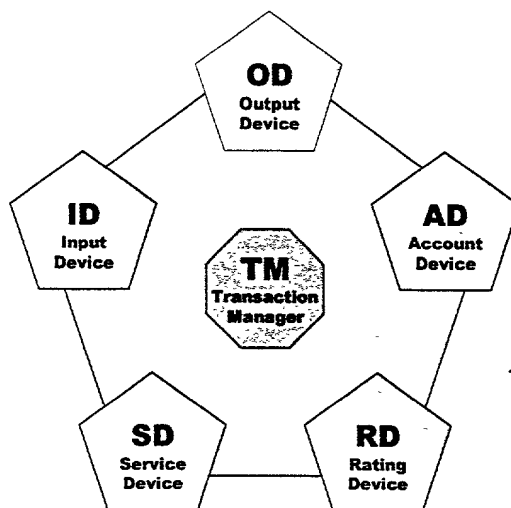
<b>Business Application</b>		<b>Business App</b> <ul style="list-style-type: none"> <li>• Financial Billing</li> <li>• Technical Support</li> <li>• Customer Service</li> </ul>	<ul style="list-style-type: none"> <li>✓ Global Data Warehouse</li> <li>✓ Workflow</li> <li>✓ 3-tier architecture</li> </ul>
<b>TxS Engine</b>		<b>Transaction Processing</b> <ul style="list-style-type: none"> <li>• Transaction Integrity</li> <li>• Roaming</li> </ul>	<ul style="list-style-type: none"> <li>✓ Vendor independent</li> <li>✓ Distributed</li> <li>✓ Scalable</li> <li>✓ Expandable</li> <li>✓ State of the Art technology</li> </ul>
<b>Device Encapsulation</b>		<b>Device implementation</b> <ul style="list-style-type: none"> <li>• Platform specific</li> <li>• Service specific</li> <li>• Placeholder for service platform integration</li> </ul>	
<b>Service Platform</b>		<b>Service Platform</b> <ul style="list-style-type: none"> <li>• e-commerce</li> <li>• Web Server</li> <li>• Telephony Debit Platform</li> </ul>	<ul style="list-style-type: none"> <li>✓ Continuity</li> <li>✓ Reuse of existing platforms</li> <li>✓ Integration of new platforms</li> <li>✓ Legacy issues respected</li> </ul>

Fig. 1



- **TM : Transaction Manager**
  - Enforces transaction integrity
- **ID : Input Device**
  - Captures end-user requests
- **SD : Service Device**
  - Provides the service
- **RD : Rating Device**
  - Rates transaction
- **AD : Account Device**
  - Manages pre-paid Accounts
- **OD : Output Device**
  - Queues TDR for Business Application

Fig. 2

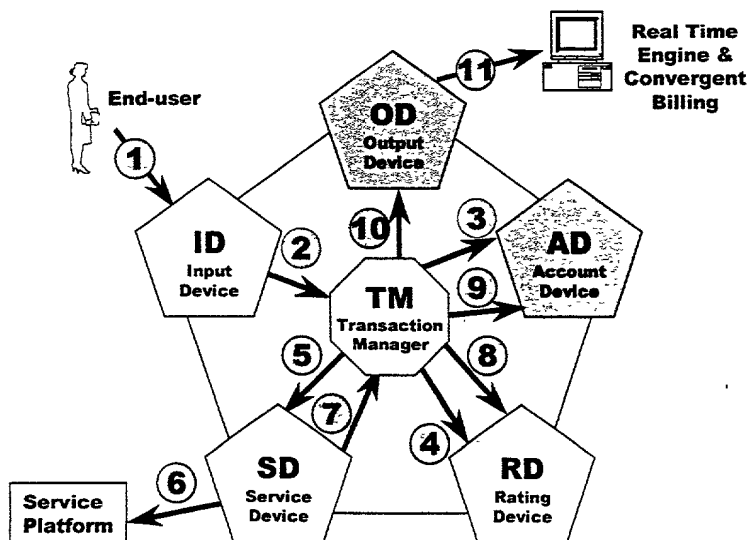


Fig. 3

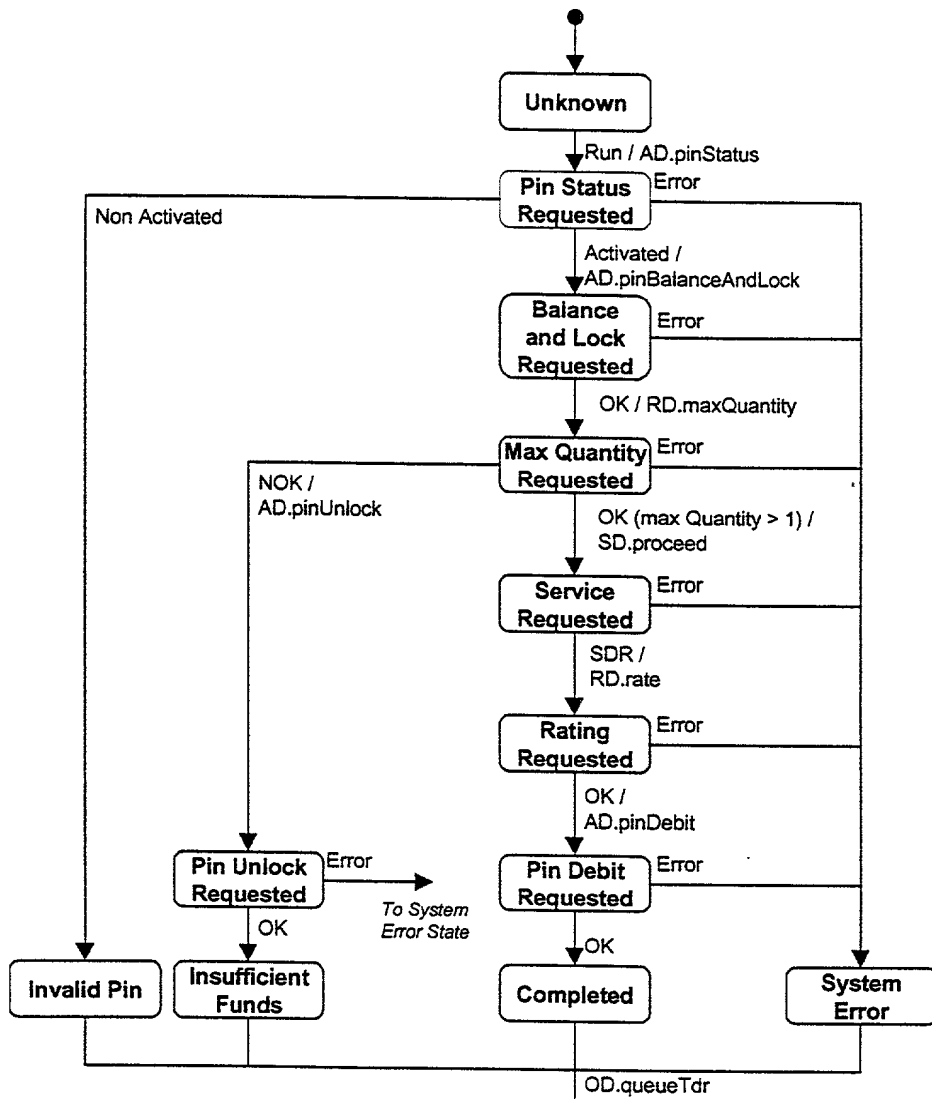
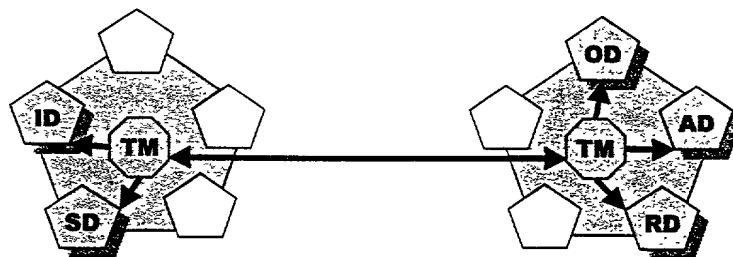


Fig. 4



**Foreign TxS**  
Captures End User Request  
Services the Request

**Home TxS**  
Account Management  
Rating  
TDR Queuing

Fig. 5

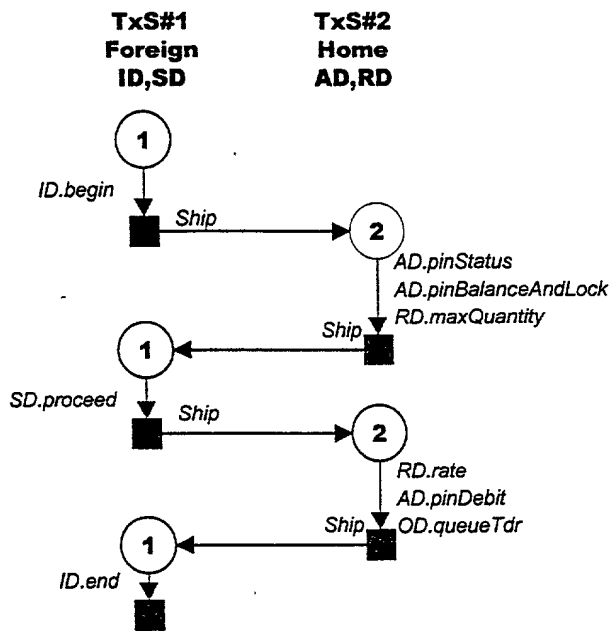


Fig. 6

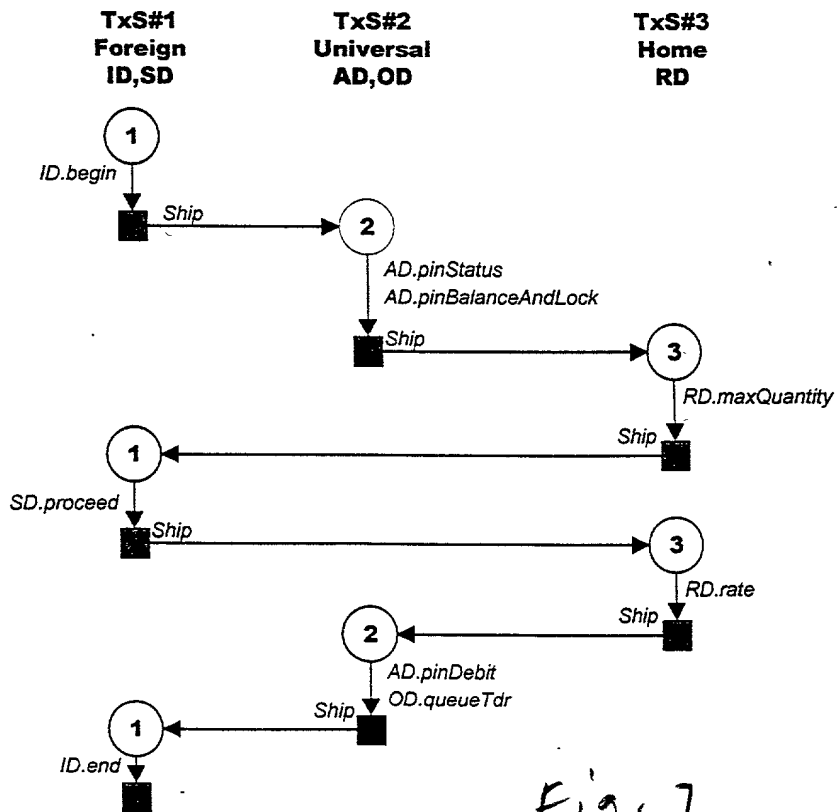


Fig. 7

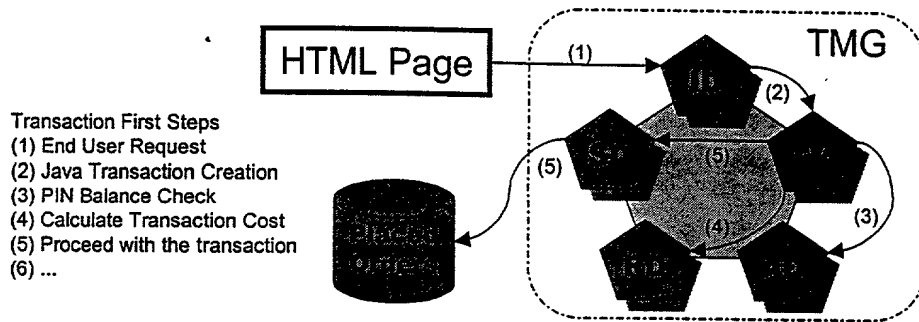


Fig. 8

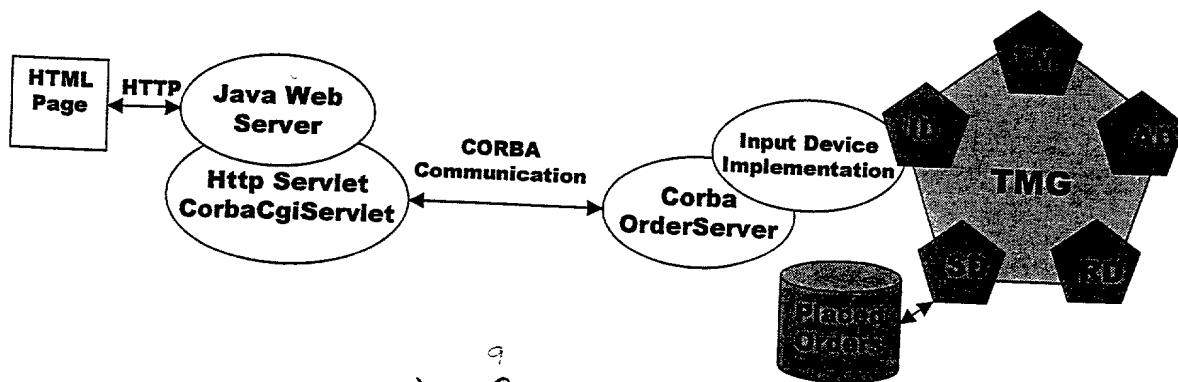


Fig 9

```
<FORM action="/servlet/CorbaCgiServlet" method="Get">
  <INPUT type=hidden name=CorbaCgiServerName value=OrderServer>
  ...
</FORM>
```

Fig. 10

```
public void doGet(HttpServletRequest req, HttpServletResponse res)
    throws ServletException, IOException
{
    String serverName = req.getParameter("CorbaCgiServerName");
    ServletOutputStream out = res.getOutputStream();
    ...
    tools.web.CorbaCgiInterface server =
        CorbaCgiInterfaceHelper.bind(tools.corba.CorbaUtil.orb(),serverName);
    ...
    String output = server.exec(req.getQueryString());
    res.setContentType("text/html");
    out.println(output);
}
```

Fig. 11

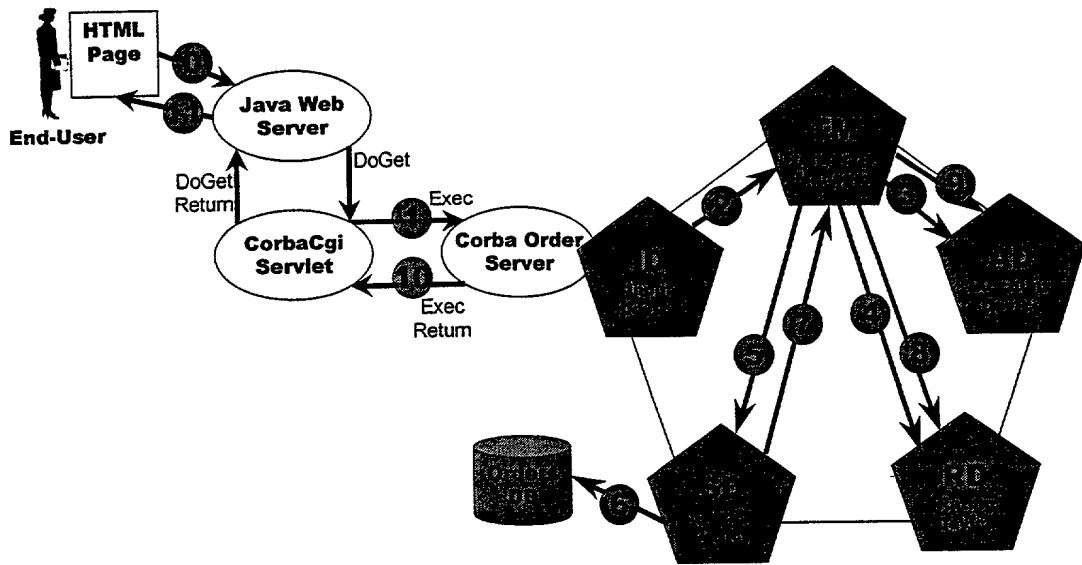


Fig. 12

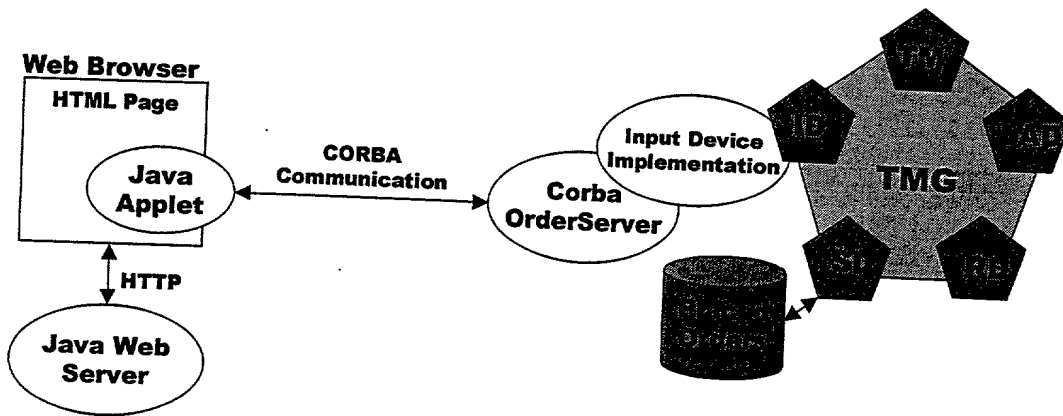


Fig. 13

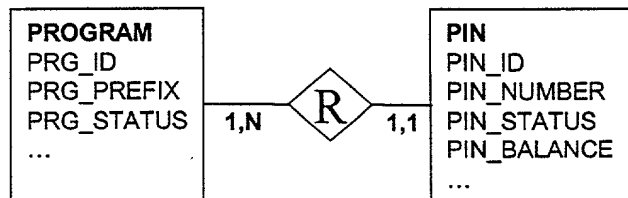


Fig. 14

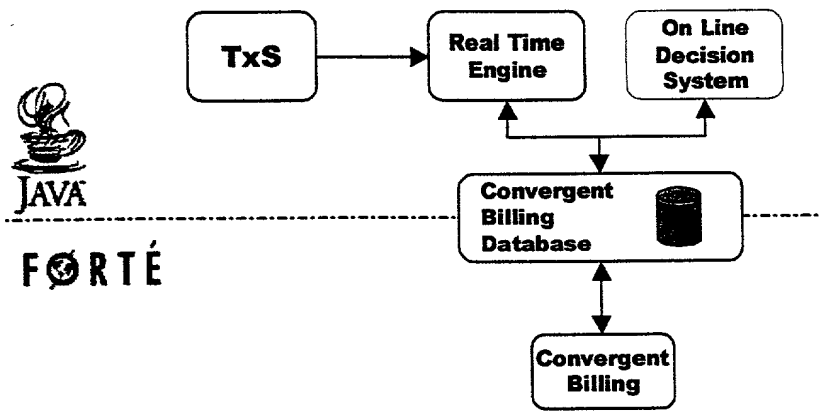


Fig. 15

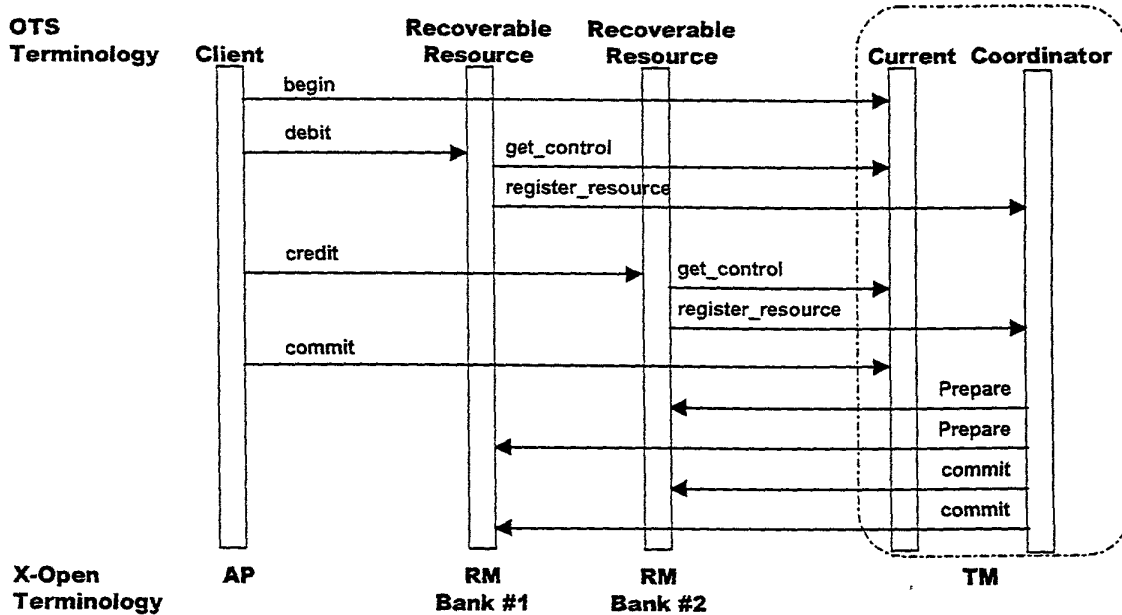


Fig. 16

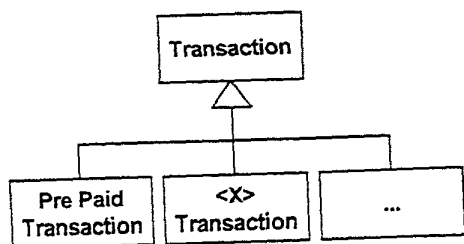


Fig. 17

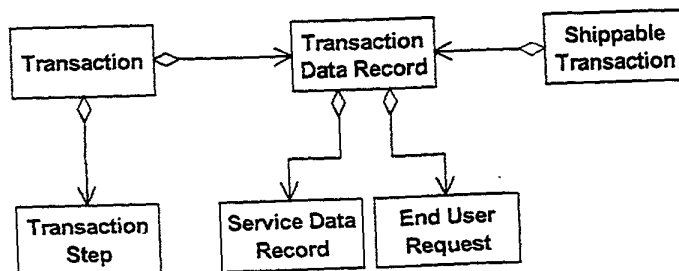


Fig. 18

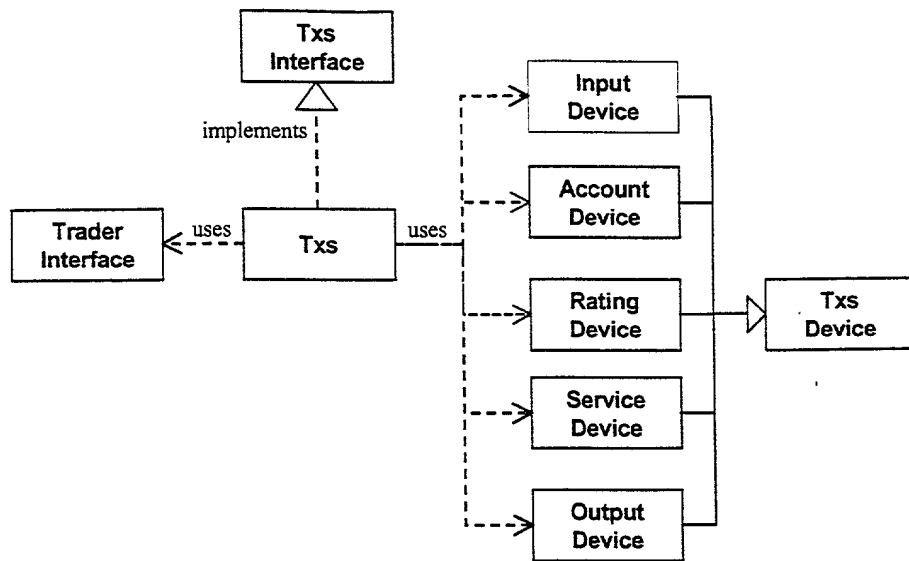


Fig. 19

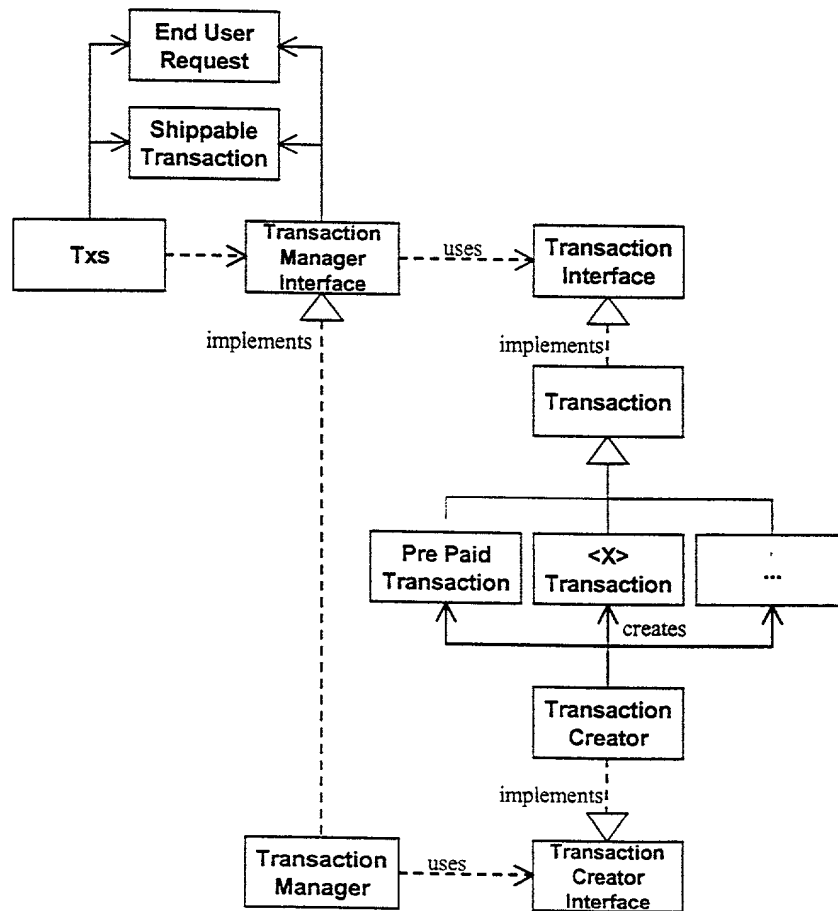


Fig. 20



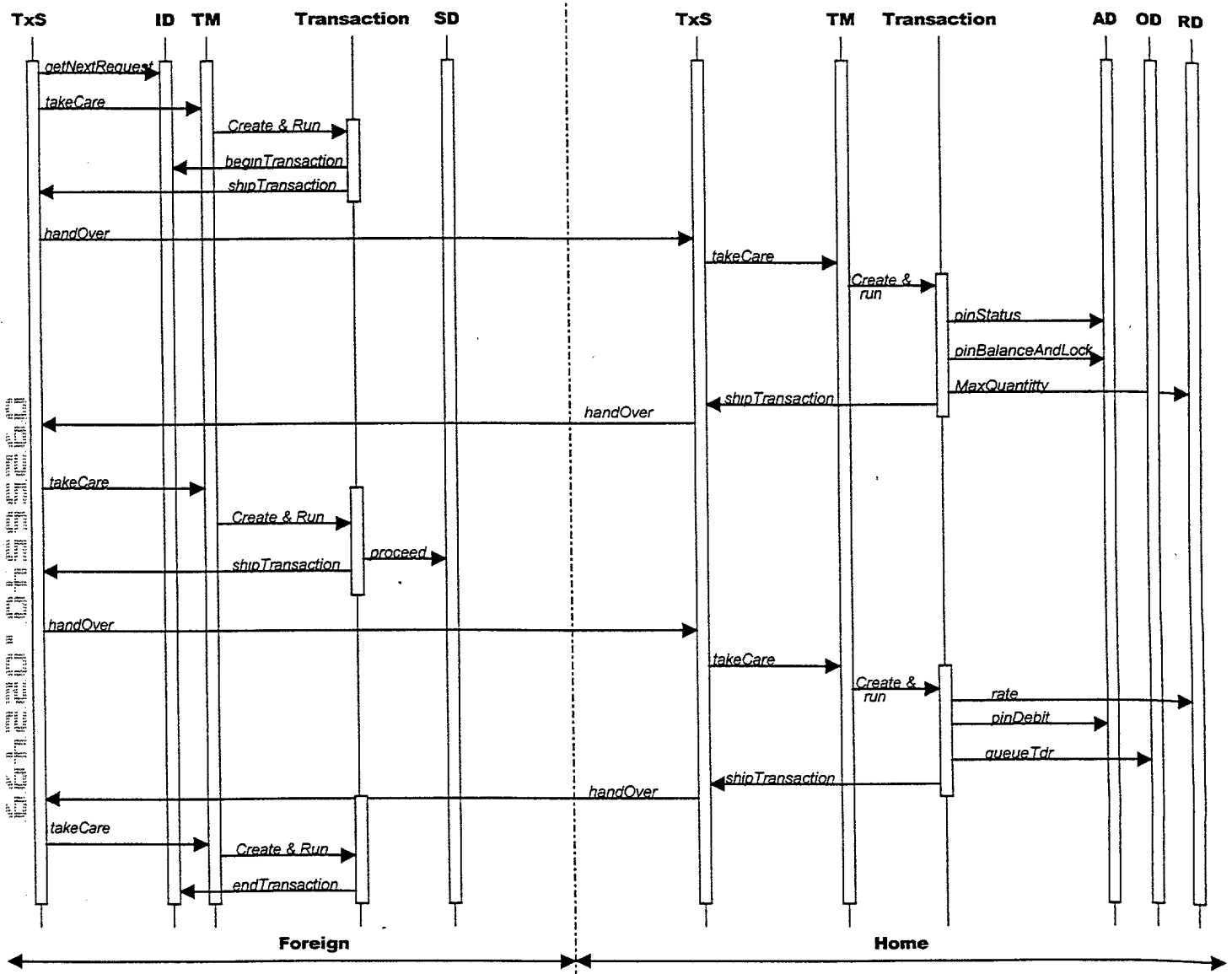


Fig. 21

**COMBINED DECLARATION, POWER OF ATTORNEY  
AND VERIFIED STATEMENT CLAIMING SMALL ENTITY STATUS**

As the below named inventor, I hereby declare my residence, post office address and citizenship are as stated below next to my name; and I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

the specification of which

- ☒ is attached hereto.  
☐ was filed on \_\_\_\_\_ as ☐ Serial No. 0 / \_\_\_\_\_  
 \_\_\_\_\_ or ☐ Express Mail No., *as Serial No. not yet known* \_\_\_\_\_  
 \_\_\_\_\_ and was amended on \_\_\_\_\_ (if  
*applicable*).  
☐ was described and claimed in PCT International Application No. \_\_\_\_\_  
 \_\_\_\_\_ filed on \_\_\_\_\_ and as amended under PCT  
 Article 19 on \_\_\_\_\_ (if any).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge to the duty to disclose information which is material to patentability as defined in 37, Code of Federal Regulations, §1.56(a). I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application on which priority is claimed:

**PRIOR FOREIGN APPLICATION(S)**

			Priority Claimed	
			Yes	No
(Number)	(Country)	(Day/month/year filed)	[ ]	[ ]

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(App. Serial No.)	(Filing date)	(Status)
		(patented, pending, abandoned)

## CLAIM FOR BENEFIT OF PRIOR U.S. PROVISIONAL APPLICATION(S)

I hereby claim the benefit under Title 35, United States Code, §119(e) of any United States provisional application(s) listed below:

PROVISIONAL APPLICATION NUMBER

FILING DATE

60/075,872

February 25, 1998

And I hereby appoint Ernest I. Gifford, Reg. No. 20,644; Allen M. Krass, Reg. No. 18,277; Irvin I. Groh, Reg. No. 17,505; Douglas W. Sprinkle, Reg. No. 27,394; Thomas E. Anderson, Reg. No. 31,318; Ronald W. Citkowski, Reg. No. 34,732; Judith M. Riley, Reg. No. 30,311; Theresa A. Orr, Reg. No. 34,890; Douglas J. McEvoy, Reg. No. 34,385; Ellen S. Cogen, Reg. No. 38,109; Roberta J. Morris, Reg. No. 33,196; John G. Posa, Reg. No. 37,424; and Douglas L. Wathen, Reg. No. 41,369 as my attorneys, to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith. Send all correspondence to:

John G. Posa, Esq.  
GIFFORD, KRASS, GROH, SPRINKLE,  
PATMORE, ANDERSON & CITKOWSKI, P.C.,  
280 N. Old Woodward Ave., Suite 400  
Birmingham, Michigan 48009  
Telephone (734) 913-9300

I hereby declare that I qualify as an independent inventor as defined in 37 CFR 1.9(c) for purposes of paying reduced fees under Section 41(a) and (b) of Title 35, United States Code, to the Patent and Trademark Office with regard to the invention described in the specification. I have not assigned, granted, conveyed or licensed and am under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who could not be classified as an independent inventor under 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a non-profit organization under 37 CFR 1.9(e). Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

- ☒ no such persons, concern, or organization  
☐ persons, concerns or organizations listed below

FULL NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

☐ Individual

☐ Small Business Concern

☐ Non-Profit Organization

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b)).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by

fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon.

**Full name of first inventor** CAROLINE BASER

**Inventor's signature** Caroline Baser

**Date** 2/19/99

**Country of Citizenship** US

**Residence** Ft. Lauderdale, Florida

**Post Office Address** 1512 S.W. 5<sup>th</sup> St., Ft. Lauderdale, Florida 33312

**Full name of second joint inventor** DONALD FLETCHER

**Inventor's signature** \_\_\_\_\_

**Date** \_\_\_\_\_

**Country of Citizenship** US

**Residence** \_\_\_\_\_

**Post Office Address** \_\_\_\_\_

**Full name of third joint inventor** OLIVER GOROSTIS

**Inventor's signature** \_\_\_\_\_

**Date** \_\_\_\_\_

**Country of Citizenship** France

**Residence** Octeville, France

**Post Office Address** 9 Rue d' Auvergne, 50 130 Octeville, France

**Full name of fourth joint inventor** FRANCOIS MAINE

**Inventor's signature** \_\_\_\_\_

**Date** \_\_\_\_\_

**Country of Citizenship** \_\_\_\_\_

**Residence** \_\_\_\_\_

**Post Office Address** \_\_\_\_\_

**Full name of fifth joint inventor** MICHAEL FREE

Inventor's signature \_\_\_\_\_

Date \_\_\_\_\_ Country of Citizenship Australia

Residence New South Wales, Australia

Post Office Address 5/4 Rose Crescent, Mosman Bay 2088, New South Wales, Australia